APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/722,630	11/26/2003	Brent Roberts	717841.5	7332

27128 7590 12/27/2007	EXAMINER	
BLACKWELL SANDERS LLP	LEE, JINHEE J	
720 OLIVE STREET		
SUITE 2400	ART UNIT	PAPER NUMBER
ST. LOUIS, MO 63101	2174	

MAIL DATE	DELIVERY MODE
12/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
	10/722,630	ROBERTS ET AL.
Office Action Summary	Examiner	Art Unit	
	Jinhee J. Lee	2174	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS,
WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>18 October 2007</u>.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-17</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-17</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 101

1.   35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of
matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the
conditions and requirements of this title.

2.   Claims 10-12 are rejected under 35 U.S.C. 101 because the claimed invention is
directed to non-statutory subject matter.

The basic of this rejection is set forth in a two-prong test of :
        (1) whether the invention is within the technology arts; and
        (2) whether the invention produces a useful, concrete, and tangible result.

        Claims 10-12 are directed to a software tool where the inputs are numbers and
the results are also numbers, and/or are directed to a computer program stored in a
computer readable storage medium for implementing the method. In order for a claimed
invention that is directed to such a computer implemented method of calculation, or a
computer program stored in a computer readable storage for implementing a
computation to be statutory, the claimed invention must accomplish a practical
application. That is the claimed invention must transform an article or physical object to
a different state or thing, or produce a useful, concrete and tangible result. State Street,
149 F.3d at 1373-74, 47 USPQ2d at 1601-02.

        It is clear from claims 10-12 that the claims merely involves calculations and
manipulations of data in performing computations. The claimed invention does not result
in a physical transformation. The inputs are numbers and the outputs are also numbers.
The result of the invention is merely numerical values without a practical application
recited in the claims. It is not real world result, and thus is not useful, concrete and

tangible. Therefore, the claimed invention is directed to non-statutory subject matter as

the claims fail to assert a practical application to the invention.

In view of the above, claims 10-12 are therefore directed to non-statutory subject

matter.

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public
use or on sale in this country, more than one year prior to the date of application for patent in the United
States.

4.   Claims 1-17 are rejected under 35 U.S.C. 102(b) as being anticipated by

Sanderson (20020101448).

Re claim 1, Sanderson discloses an integrated system of frameworks and data

repositories for generating a graphical user interface in a client-server environment

comprising: a user interface (UI) repository (data factory 210 for example) residing in a

database accessible to a client server hardware system, where said UI repository,

contains a UI element (data element 208 for example), which defines data element

attributes including data type, how to display data and labels (see figure 2 for example);

a screen repository (specification 209 for example) residing in the database storage

device accessible to the client-server hardware system, where said screen repository

includes screen attributes, which defines the hierarchical navigational tree structure of

screens for an graphical user interface (GUI) application and further defines what

screen will be constructed and defines a GUI component of the screen based on the

data type (workflow data, configuration data, see abstract for example); a data binding

framework (content factory 213 for example) operable to bind data to the UI element

and the GUI component based on data type (inherent in order to display); a (GUI)

framework (controller 201c for example) operably residing at a client in the client-server

hardware system, where said (GUI) framework is operable to control how data is

handled and processed within the GUI component of the GUI application based on data

type including binding data to the GUI component utilizing the data binding framework;

and

  a navigation framework (view 201b, model 201a for example) operably residing

at the client, where said navigation framework controls generating and displaying of the

screens within an application and further builds a navigation tree structure based on the

screen attributes; and

  an object oriented software application (see paragraph 0037, object-oriented

implementation and XML, see paragraph 0046 for example) for generating a graphical

user interface being executed in said client-server hardware system (client 201D, server

206 in figure 2 for example) having functional interfaces for accessing repositories and

frameworks accessible by the client-server network (see figure 2 for example).

  Re claim 2, Sanderson discloses an integrated system of frameworks and data

repositories for generating a graphical user interface, further comprising: a security

framework (using 442 for example, inherent) operable to communicate information to

the navigation framework causing the navigation framework to selectively deny a user

access to screens by not providing the user with selections that would navigate to the

screen (see figure 4b).

Re claim 3, Sanderson discloses an integrated system of frameworks and data

repositories for generating a graphical user interface, further comprising: a verification

framework (validator 211 for example) operable to apply business rules to data

contained in a data set and determine if the data is in error and if in error the verification

framework communicates with the GUI component to display an error message.

Re claim 4, Sanderson discloses an integrated system of frameworks and data

repositories for generating a graphical user interface in a client-server environment

comprising:

a graphical user interface (GUI) framework (201c for example) operably residing

at a client server hardware system in a client-server network environment, where said

GUI framework is operable to control how data is handled and processed within a GUI

component of a GUI application based on a data type including binding data to the GUI

component utilizing a data binding framework;

a collection of integrated repositories (210 for example) relationally inter

referenced by UI elements defining the data type within their respective attribute tables

operable for accessing and integrating all attribute elements relating to generating a

graphical user interface;

a collection of executable object oriented routines (209 for example) being

executed on the client-server hardware system (client 201D, server 206, see figure 2 for

example) and operable to manipulate the GUI framework;

a collection of XML files (content 207 for example) operable to access and export data from the repositories at run time for use by the GUI application; an XML layout manager (format 212 for example) operable to define the screen layout from the collection of XML files; and a navigation framework operably residing at the client, where said navigation framework (view 201b for example) controls generating and displaying of screens within the GUI application based upon the XML lay out manager and the repository attributes accessed and exported by the collection of XML files and further builds the navigation tree structure based on the repository attributes.

Re claim 5, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a security framework (using 442, inherent) operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen (see figure 4b).

Re claim 6, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a verification framework (validator 211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 7, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface in a client-server environment comprising:

a screen repository (209 for example) residing in a database storage device accessible by a client-server hardware system, where said screen repository includes screen attributes, which defines the hierarchical navigational tree structure of screens for an graphical user interface (GUI) application and further defines what screen will be constructed and defines a GUI components of the screen;

a user interface (UI) repository (210 for example) residing in the database storage device accessible to the client server hardware system, where said UI repository, contains a UI element, which defines data element attributes including data type, how to display data and labels; and

a data binding framework (213 for example) operable to bind data to the UI element and the GUI component based on the data type defined in the UI repository; and

an object oriented software application (see paragraph 0037 object oriented implementation and XML in paragraph 0046 for example) for generating a graphical user interface being executed in said client-server hardware system (client 201D, server 206, see figure 2 for example) having functional interfaces for accessing repositories and frameworks accessible by the client-server hardware system (see figure 2 for example).

Re claim 8, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a security framework (using 442, inherent) operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to

screens by not providing the user with selections that would navigate to the screen (see figure 4b for example).

Re claim 9, Sanderson discloses an integrated system of frameworks and data repositories for generating a graphical user interface, further comprising: a verification framework (211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 10, Sanderson discloses an integrated system tool comprising:

an administrative computing tool application being executed on a computer (server 206 for example) including, a navigation tool (201b for example) for building a navigation framework adapted to control the generation of screens for a graphical user interface (GUI) application and further adapted to define the hierarchical relationship of the screens, a screen repository tool (209 for example) operable to build a repository of screen attributes to establish a hierarchical screen navigation structure and a corresponding Java class construct to be executed and a GUI component, a user interface repository tool (210 for example) operable to build a user interface repository having user interface attribute tables of user interface elements corresponding to the GUI component, and a data binding framework tool (213 for example) operable to build a data binding framework operable to bind data from an appropriate data set to the user interface element and the GUI component.

Re claim 11, Sanderson discloses an integrated system tool, where the administrator computing tool further comprises: a security framework tool (using 442 for

example) operable to build a security framework operable to communicate information
to the navigation framework causing the navigation framework to selectively deny a user
access to screens by not providing the user with selections that would navigate to the
screen (see figure 4b).

Re claim 12, Sanderson discloses an integrated system tool, where the
administrator computing tool further comprises: a verification framework tool (211 for
example) operable to build a verification framework operable to apply business rules to
data contained in a data set and determine if the data is in error and if in error the
verification framework communicates with the GUI component to display an error
message.

Re claim 13, Sanderson discloses a method of generating a graphical user
interface utilizing an integrated system of frameworks and data repositories comprising
the steps of: Receiving a screen (201d to 207 for example) request to a graphical user
interface (GUI) application based on a user input; Accessing and constructing (from
201b to 201a for example) a basic screen and screen attributes from a screen
repository corresponding to the user input as determined by a navigation framework;
Binding GUI components (create instance in 210 for example) defined by the screen
attributes with user interface elements from a UI repository based on a data type
defined in the UI repository (see figure 2 for example); Binding data to the GUI
components and UI elements based on the data type (format 212 for example); and
Displaying the screen on a display system (201b to 201a for example).

Re claim 14, Sanderson discloses a method of generating a graphical user interface, further comprising: filtering a screen (with 442 for example) with a security framework operable to communicate information to the navigation framework causing the navigation framework to selectively deny a user access to screens by not providing the user with selections that would navigate to the screen.

Re claim 15, Sanderson discloses a method of generating a graphical user interface as recited in claim 13, further comprising: displaying an error message (using 211 for example) with a verification framework operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Re claim 16, Sanderson discloses a method for building an integrated system of frameworks and data repositories for generating a graphical user interface comprising the steps of: building a graphical user interface (GUI) framework (201c for example) operable to reside at a client in a client-server network environment, where said GUI framework is operable to control how data is handled and processed within a GUI component of a GUI application including binding data to the GUI component utilizing a data binding framework; and

building a collection (with 210 for example) of integrated repositories to be relationally inter referenced by UI element data type within their respective attribute tables operable for accessing and integrating all attribute elements relating to generating a graphical user interface; and displaying a graphical user interface (see figure 2 and abstract for example).

Re claims 17, Sanderson discloses a method for building an integrated system further comprising the steps of: building a verification framework (with 211 for example) operable to apply business rules to data contained in a data set and determine if the data is in error and if in error the verification framework communicates with the GUI component to display an error message.

Response to Arguments

5.   Applicant's arguments filed 7/18/07 have been fully considered but they are not persuasive.

Regarding the 101 rejection, claims 10-12 still do not meet the statutory requirements under 35 U.S.C. 101, lacking a physical article to realize the functionality. Furthermore, regarding the method claims, the tangible result of the functional relations are lacking, and therefore, lacking the realization of the method steps.

Regarding the 102 rejection:

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., "Integrated frameworks that revolve around data types where various applications can utilize the frameworks while maintaining commonality with regard to presentation, user interaction, and functionality" and "a system of integrated frameworks that revolve around data type whereby the data type of a UI element determines how data is bound, how data is controlled and handled in the GUI component of a screen within an overall navigation framework.") are not recited in the rejected claim(s). Although the claims are

interpreted in light of the specification, limitations from the specification are not read into

the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Furthermore, UI repository with UI element can be interpreted as data factory

210, which has data elements 208 for the user interface (UI). Data elements for user

interface (UI) meet the limitation of "UI element". A data can be a user interface

element. Examiner disagrees with the applicant's arguments.

In response to applicant's arguments that the prior art does not disclose "a data

binding framework", examiner disagrees. Item 213 of Sanderson discloses

ContentFactory where the content of the user interface is created. This is providing

data binding framework, where the data gathered or determined is created or binded to

each other.

In response to applicant's arguments that the prior art does not disclose

"navigation framework", examiner disagrees. Claim states that "said navigation

framework controls generating and displaying of the screens within an application"

which is the function of item 204B View in prior art. Further, figure 1 shows in the view

"navigation tree structure based on the screen attributes.

Conclusion

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Jinhee J. Lee whose telephone number is 571-272-

1977. The examiner can normally be reached on M-F at 8:30AM-5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, David Wiley can be reached on 571-272-2100 ext. 74. The fax phone

number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jinhee J Lee/
Primary Examiner, Art Unit 2174

jjl